WIKIPEDIA
The Free Encyclopedia

File Transfer Protocol

The **File Transfer Protocol** (**FTP**) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture using separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a plain-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

File Transfer Protocol

	Communication protocol
Purpose	File transfer
Developer(s)	Abhay Bhushan for RFC 114
Introduction	April 16, 1971
OSI layer	Application layer
Port(s)	21 for control, 20 for data transfer
RFC(s)	RFC 959

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many dedicated FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications such as HTML editors and file managers.

An FTP client used to be commonly integrated in web browsers, where file servers are browsed with the URI prefix "[ftp://"FTP link"]. *FTP server* (FTP). {{cite web}}: Check |url= value (help) (To view documents see Help:FTP.)". In 2021, FTP support was dropped by Google Chrome and Firefox,[4][5] two major web browser vendors, due to it being superseded by the more secure SFTP and FTPS; although neither of them have implemented the newer protocols.[6][7]

Internet history timeline

Early research and development:

- 1960–1964: RAND networking concepts developed
- 1962–1964: ARPA networking ideas
- 1965: NPL network concepts conceived
- 1966: Merit Network founded
- 1967: ARPANET planning begins
- 1967: Symposium on Operating Systems Principles
- 1969: NPL followed by the ARPANET carry their first packets
- 1970: Network Information Center (NIC)
- 1971: Tymnet switched-circuit network
- 1972: Merit Network's packet-switched network operational
- 1972: Internet Assigned Numbers Authority (IANA) established

History of FTP servers

The original specification for the File Transfer Protocol was written by Abhay Bhushan and published as RFC 114 (https://www.rfc-editor.org/rfc/rfc114) on 16 April 1971. Until 1980, FTP ran on NCP, the predecessor of TCP/IP.[2] The protocol was later replaced by a TCP/IP version, RFC 765 (https://www.rfc-editor.org/rfc/rfc765)

(June 1980) and RFC 959 (https://www.rfc-editor.org/rfc/rfc959) (October 1985), the current specification. Several proposed standards amend RFC 959 (https://www.rfc-editor.org/rfc/rfc959), for example RFC 1579 (https://www.rfc-editor.org/rfc/rfc1579) (February 1994) enables Firewall-Friendly FTP (passive mode), RFC 2228 (https://www.rfc-editor.org/rfc/rfc2228) (June 1997) proposes security extensions, RFC 2428 (https://www.rfc-editor.org/rfc/rfc2428) (September 1998) adds support for IPv6 and defines a new type of passive mode.[8]

Protocol overview

Communication and data transfer

FTP may run in *active* or *passive* mode, which determines how the data connection is established.[9] (This sense of "mode" is different from that of the MODE command in the FTP protocol.)

- In active mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.
- In situations where the client is behind a firewall and unable to accept incoming TCP connections, *passive mode* may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server,[9] which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received.[10]

Both modes were updated in September 1998 to support IPv6. Further changes were introduced to the passive mode at that time, updating it to *extended passive mode*.[11]

The server responds over the control connection with three-digit status codes in ASCII with an optional text message. For example, "200" (or "200 OK") means that the last command was successful. The numbers represent the code for the response and the optional text represents a human-readable explanation or request (e.g. <Need account for storing file>).[1] An ongoing transfer of file data over the data connection can be aborted using an interrupt message sent over the control connection.

FTP needs two ports (one for sending and one for receiving) because it was originally designed to operate on top of Network Control Protocol (NCP), which was a simplex protocol that utilized two port

- 1973: CYCLADES network demonstrated
- 1973: PARC Universal Packet development begins
- 1974: Transmission Control Program specification published
- 1975: Telenet commercial packet-switched network
- 1976: X.25 protocol approved and deployed on public data networks
- 1978: Minitel introduced
- 1979: Internet Activities Board (IAB)
- 1980: USENET news using UUCP
- 1980: Ethernet standard introduced
- 1981: BITNET established

Merging the networks and creating the Internet:

- 1981: Computer Science Network (CSNET)
- 1982: TCP/IP protocol suite formalized
- 1982: Simple Mail Transfer Protocol (SMTP)
- 1983: Domain Name System (DNS)
- 1983: MILNET split off from ARPANET
- 1984: OSI Reference Model released
- 1985: First .COM domain name registered
- 1986: NSFNET with 56 kbit/s links
- 1986: Internet Engineering Task Force (IETF)
- 1987: UUNET founded
- 1988: NSFNET upgraded to 1.5 Mbit/s (T1)

addresses, establishing two connections, for two-way communications. An odd and an even port were reserved for each application layer application or protocol. The standardization of TCP and UDP reduced the need for the use of two simplex ports for each application down to one duplex port,[12]:15 but the FTP protocol was never altered to only use one port, and continued using two for backwards compatibility.

NAT and firewall traversal

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both NATs and firewalls, which do not allow connections from the Internet towards internal hosts.[13] For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to solve this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server.[13] This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an application-level gateway for this purpose.[13]

Data types

While transferring data over the network, five data types are defined:[2][3][8]

- ASCII (TYPE A): Used for text. Data is converted, if needed, from the sending host's character representation to "8-bit ASCII" before transmission, and (again, if necessary) to the receiving host's character representation, including newlines. As a consequence, this mode is inappropriate for files that contain data other than ASCII.
- Image (TYPE I, commonly called Binary mode): The sending machine sends each file byte by byte, and the recipient stores the bytestream as it receives it. (Image mode support has been recommended for all implementations of FTP).
- EBCDIC (TYPE E): Used for plain text between hosts using the EBCDIC character set.
- Local (TYPE L *n*): Designed to support file transfer between machines which do not use 8-bit bytes, e.g. 36-bit systems such as DEC PDP-10s. For example, "TYPE L 9" would be used to transfer data in 9-bit bytes, or "TYPE L

- 1988: Morris worm
- 1988: Complete Internet protocol suite
- 1989: Border Gateway Protocol (BGP)
- 1989: PSINet founded, allows commercial traffic
- 1989: Federal Internet Exchanges (FIX East|FIXes)
- 1990: GOSIP (without TCP/IP)
- 1990: ARPANET decommissioned
- 1990: Advanced Network and Services (ANS)
- 1990: UUNET/Alternet allows commercial traffic
- 1990: Archie search engine
- 1991: Wide area information server (WAIS)
- 1991: Gopher
- 1991: Commercial Internet eXchange (CIX)
- 1991: ANS CO+RE allows commercial traffic
- 1991: World Wide Web (WWW)
- 1992: NSFNET upgraded to 45 Mbit/s (T3)
- 1992: Internet Society (ISOC) established
- 1993: Classless Inter-Domain Routing (CIDR)
- 1993: InterNIC established
- 1993: AOL added USENET access
- 1993: Mosaic web browser released
- 1994: Full text web search engines
- 1994: North American Network Operators' Group (NANOG) established

36" to transfer 36-bit words. Most contemporary FTP clients/servers only support L 8, which is equivalent to I.
- Unicode text files using UTF-8 (TYPE U): defined in an expired Internet Draft[14] which never became an RFC, though it has been implemented by several FTP clients/servers.

Note these data types are commonly called "modes", although ambiguously that word is also used to refer to active-vs-passive communication mode (see above), and the modes set by the FTP protocol MODE command (see below).

For text files (TYPE A and TYPE E), three different format control options are provided, to control how the file would be printed:

- Non-print (TYPE A N and TYPE E N) – the file does not contain any carriage control characters intended for a printer
- Telnet (TYPE A T and TYPE E T) – the file contains Telnet (or in other words, ASCII C0) carriage control characters (CR, LF, etc)
- ASA (TYPE A A and TYPE E A) – the file contains ASA carriage control characters

These formats were mainly relevant to line printers; most contemporary FTP clients/servers only support the default format control of N.

File structures

File organization is specified using the STRU command. The following file structures are defined in section 3.1.1 of RFC959:

- **F** or FILE structure (stream-oriented). Files are viewed as an arbitrary sequence of bytes, characters or words. This is the usual file structure on Unix systems and other systems such as CP/M, MS-DOS and Microsoft Windows. (Section 3.1.1.1)
- **R** or RECORD structure (record-oriented). Files are viewed as divided into records, which may be fixed or variable length. This file organization is common on mainframe and midrange systems, such as MVS, VM/CMS, OS/400 and VMS, which support record-oriented filesystems.
- **P** or PAGE structure (page-oriented). Files are divided into pages, which may either contain data or metadata; each page may also have a header giving various attributes. This file structure was specifically designed for TENEX systems, and is generally not supported on other platforms. RFC1123 section 4.1.2.3 recommends that this structure not be implemented.

Commercialization, privatization, broader access leads to the modern Internet:

- 1995: New Internet architecture with commercial ISPs connected at NAPs
- 1995: NSFNET decommissioned
- 1995: GOSIP updated to allow TCP/IP
- 1995: very high-speed Backbone Network Service (vBNS)
- 1995: IPv6 proposed
- 1996: AOL changes pricing model from hourly to monthly
- 1998: Internet Corporation for Assigned Names and Numbers (ICANN)
- 1999: IEEE 802.11b wireless networking
- 1999: Internet2/Abilene Network
- 1999: vBNS+ allows broader access
- 2000: Dot-com bubble bursts
- 2001: New top-level domain names activated
- 2001: Code Red I, Code Red II, and Nimda worms
- 2003: UN World Summit on the Information Society (WSIS) phase I
- 2003: National LambdaRail founded
- 2004: UN Working Group on Internet Governance (WGIG)
- 2005: UN WSIS phase II
- 2006: First meeting of the Internet Governance Forum

Most contemporary FTP clients and servers only support STRU F. STRU R is still in use in mainframe and minicomputer file transfer applications.

Data transfer modes

Data transfer can be done in any of three modes:[1][2]

- Stream mode (MODE S): Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.
- Block mode (MODE B): Designed primarily for transferring record-oriented files (STRU R), although can also be used to transfer stream-oriented (STRU F) text files. FTP puts each record (or line) of data into several blocks (block header, byte count, and data field) and then passes it on to TCP.[8]
- Compressed mode (MODE C): Extends MODE B with data compression using run-length encoding.

Most contemporary FTP clients and servers do not implement MODE B or MODE C; FTP clients and servers for mainframe and minicomputer operating systems are the exception to that.

Some FTP software also implements a DEFLATE-based compressed mode, sometimes called "Mode Z" after the command that enables it. This mode was described in an Internet Draft, but not standardized.[15]

GridFTP defines additional modes, MODE E[16] and MODE X,[17] as extensions of MODE B.

Additional commands

More recent implementations of FTP support the *Modify Fact: Modification Time* (MFMT) command, which allows a client to adjust that file attribute remotely, enabling the preservation of that attribute when uploading files.[18][19]

To retrieve a remote file timestamp, there's *MDTM* command. Some servers (and clients) support nonstandard syntax of the *MDTM* command with two arguments, that works the same way as *MFMT*[20]

- 2010: First internationalized country code top-level domains registered
- 2012: ICANN begins accepting applications for new generic top-level domain names
- 2013: Montevideo Statement on the Future of Internet Cooperation
- 2014: NetMundial international Internet governance proposal
- 2016: ICANN contract with U.S. Dept. of Commerce ends, IANA oversight passes to the global Internet community on October 1st

Examples of Internet services:

- 1989: AOL dial-up service provider, email, instant messaging, and web browser
- 1990: IMDb Internet movie database
- 1994: Yahoo! web directory
- 1995: Amazon online retailer
- 1995: eBay online auction and shopping
- 1995: Craigslist classified advertisements
- 1995: AltaVista search engine
- 1996: Outlook (formerly Hotmail) free web-based e-mail
- 1996: RankDex search engine
- 1997: Google Search
- 1997: Babel Fish automatic translation
- 1998: Yahoo Groups (formerly Yahoo! Clubs)

Login

FTP login uses normal username and password scheme for granting access.[2] The username is sent to the server using the USER command, and the password is sent using the PASS command.[2] This sequence is unencrypted "on the wire", so may be vulnerable to a network sniffing attack.[21] If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence.[2] If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.[2]

Anonymous FTP

A host that provides an FTP service may provide anonymous FTP access.[2] Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their email address instead of a password,[3] no verification is actually performed on the supplied data.[22] Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.[3]

Software support

File managers

Many file managers tend to have FTP access implemented, such as File Explorer (formerly Windows Explorer) on Microsoft Windows. This client is only recommended for small file transfers from a server, due to limitations compared to dedicated client software.[23] It does not support SFTP.[24]

Both the native file managers for KDE on Linux (Dolphin and Konqueror) support FTP as well as SFTP.[25][26]

On Android, the My Files file manager on Samsung Galaxy has a built-in FTP and SFTP client.[27]

Web browser

For a long time, most common web browsers were able to retrieve files hosted on FTP servers, although not all of them had support for protocol extensions such as FTPS.[3][28] When an FTP—rather than

- 1998: PayPal Internet payment system
- 1998: Rotten Tomatoes review aggregator
- 1999: 2ch Anonymous textboard
- 1999: i-mode mobile internet service
- 1999: Napster peer-to-peer file sharing
- 2000: Baidu search engine
- 2001: 2chan Anonymous imageboard
- 2001: BitTorrent peer-to-peer file sharing
- 2001: Wikipedia, the free encyclopedia
- 2003: LinkedIn business networking
- 2003: Myspace social networking site
- 2003: Skype Internet voice calls
- 2003: iTunes Store
- 2003: 4chan Anonymous imageboard
- 2003: The Pirate Bay, torrent file host
- 2004: Facebook social networking site
- 2004: Podcast media file series
- 2004: Flickr image hosting
- 2005: YouTube video sharing
- 2005: Reddit link voting
- 2005: Google Earth virtual globe
- 2006: Twitter microblogging
- 2007: WikiLeaks anonymous news and information leaks
- 2007: Google Street View
- 2007: Kindle, e-reader and virtual bookshop

an HTTP—URL is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other web content.

Google Chrome removed FTP support entirely in Chrome 88, also affecting other Chromium-based browsers such as Microsoft Edge.[29] Firefox 88 disabled FTP support by default, with Firefox 90 dropping support entirely.[30][4]

FireFTP is a discontinued browser extension that was designed as a full-featured FTP client to be run within Firefox, but when Firefox dropped support for FTP the extension developer recommended using Waterfox.[31] Some browsers, such as the text-based Lynx, still support FTP.[32]

Syntax

FTP URL syntax is described in RFC 1738 (https://www.rfc-editor.org/rfc/rfc1738), taking the form: `"FTP link" (ftp://%5buser%5b:password)`. *`user:password`*. `{{cite web}}: Check |url= value (help)]@]host[:port]/[url-path]` (the bracketed parts are optional).

For example, the URL {{Cite FTP |url=ftp://public.ftp-servers.example.com/mydirectory/myfile.txt |server=public.ftp-servers.example.com |url-status=dead |title=FTP link }} represents the file *myfile.txt* from the directory *mydirectory* on the server *public.ftp-servers.example.com* as an FTP resource. The URL {{Cite FTP |url=ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt |server=private.ftp-servers.example.com |url-status=dead |title=FTP link }} adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation (e.g., Firefox[33] and Internet Explorer[34]). By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Some variation has existed in how different browsers treat path resolution in cases where there is a non-root home directory for a user.[35]

- 2008: Amazon Elastic Compute Cloud (EC2)
- 2008: Dropbox cloud-based file hosting
- 2008: Encyclopedia of Life, a collaborative encyclopedia intended to document all living species
- 2008: Spotify, a DRM-based music streaming service
- 2009: Bing search engine
- 2009: Google Docs, Web-based word processor, spreadsheet, presentation, form, and data storage service
- 2009: Kickstarter, a threshold pledge system
- 2009: Bitcoin, a digital currency
- 2010: Instagram, photo sharing and social networking
- 2011: Google+, social networking
- 2011: Snapchat, photo sharing
- 2012: Coursera, massive open online courses
- 2016: TikTok, video sharing and social networking

Download manager

Most common download managers can receive files hosted on FTP servers, while some of them also give the interface to retrieve the files hosted on FTP servers. DownloadStudio allows not only download a file from FTP server but also view the list of files on a FTP server.[36]

Other

LibreOffice declared its FTP support deprecated from 7.4 release, this was later removed in 24.2 release.[37][38]

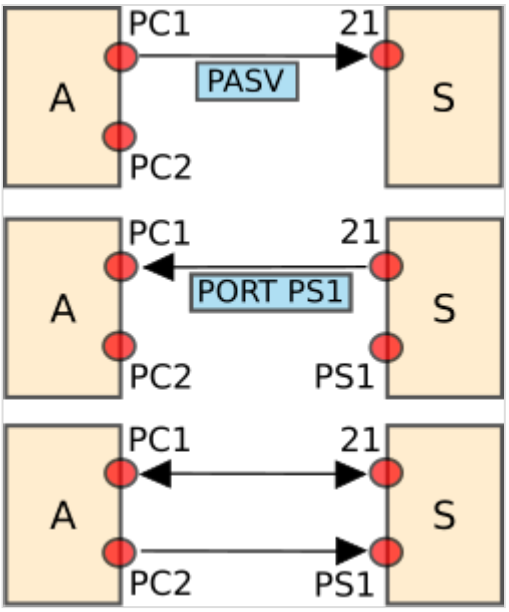


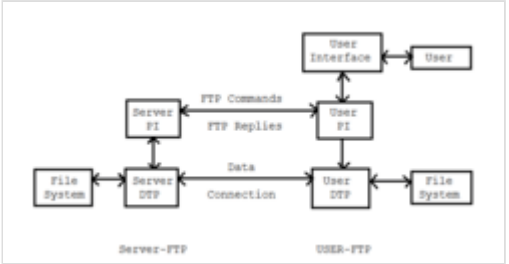
Illustration of starting a passive connection using port 21

Security

FTP was not designed to be a secure protocol, and has many security weaknesses.[39] In May 1999, the authors of RFC 2577 (https://www.rfc-editor.org/rfc/rfc2577) listed a vulnerability to the following problems:

- Brute-force attack
- FTP bounce attack
- Packet capture
- Port stealing (guessing the next open port and usurping a legitimate connection)
- Spoofing attack
- Username enumeration
- DoS or DDoS

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture (sniffing) on the network.[2][39] This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP and IMAP) that were designed prior to the creation of encryption mechanisms such as TLS or SSL.[8]



A model chart of how FTP works

Common solutions to this problem include:

1. Using the secure versions of the insecure protocols, e.g., FTPS instead of FTP and TelnetS instead of Telnet.
2. Using a different, more secure protocol that can handle the job, e.g. SSH File Transfer Protocol or Secure Copy Protocol.
3. Using a secure tunnel such as Secure Shell (SSH) or virtual private network (VPN).

A computer at Amundsen–Scott South Pole Station logging into an FTP server and transferring a file, in 1994

FTP over SSH

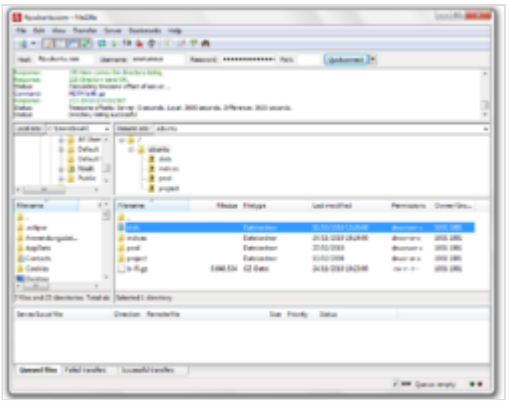
FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection.[39] Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no confidentiality or integrity protection.

Primitive FTPd on Android, actively running an FTP and SFTP server

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwardings for FTP data channels. Software packages that support this mode include:

- Tectia ConnectSecure (Win/Linux/Unix)[40] of SSH Communications Security's software suite

FTP over SSH should not be confused with SSH File Transfer Protocol (SFTP).



FileZilla client running on Windows, one of the best known FTP client software

Derivatives

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request FTP sessions to be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in RFC 4217 (https://www.rfc-editor.org/rfc/rfc4217). Implicit FTPS is an outdated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

SSH File Transfer Protocol

The SSH file transfer protocol (chronologically the second of the two protocols abbreviated SFTP) transfers files and has a similar command set for users, but uses the Secure Shell protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software, though some FTP client software offers support for the SSH file transfer protocol as well.

Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) is a simple, lock-step FTP that allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of booting from a local area network, because TFTP is very simple to implement. TFTP lacks security and most of the advanced features offered by more robust file transfer protocols such as File Transfer Protocol. TFTP was first standardized in 1981 and the current specification for the protocol can be found in RFC 1350 (https://www.rfc-editor.org/rfc/rfc1350).

Simple File Transfer Protocol

Simple File Transfer Protocol (the first protocol abbreviated SFTP), as defined by RFC 913 (https://www.rfc-editor.org/rfc/rfc913), was proposed as an (unsecured) file transfer protocol with a level of complexity intermediate between TFTP and FTP. It was never widely accepted on the Internet, and is now assigned Historic status by the IETF. It runs through port 115, and often receives the initialism of *SFTP*. It has a command set of 11 commands and support three types of data transmission: ASCII, binary and continuous. For systems with a word size that is a multiple of 8 bits, the implementation of binary and continuous is the same. The protocol also supports login with user ID and password, hierarchical folders and file management (including *rename*, *delete*, *upload*, *download*, *download with overwrite*, and *download with append*).

FTP commands

FTP reply codes

Below is a summary of FTP reply codes that may be returned by an FTP server. These codes have been standardized in RFC 959 (https://www.rfc-editor.org/rfc/rfc959) by the IETF. The reply code is a three-digit value. The first digit is used to indicate one of three possible outcomes — success, failure, or to indicate an error or incomplete reply:

- 2yz – Success reply
- 4yz or 5yz – Failure reply
- 1yz or 3yz – Error or Incomplete reply

The second digit defines the kind of error:

- x0z – Syntax. These replies refer to syntax errors.
- x1z – Information. Replies to requests for information.
- x2z – Connections. Replies referring to the control and data connections.
- x3z – Authentication and accounting. Replies for the login process and accounting procedures.
- x4z – Not defined.
- x5z – File system. These replies relay status codes from the server file system.

The third digit of the reply code is used to provide additional detail for each of the categories defined by the second digit.

See also

- Comparison of FTP client software
- Comparison of FTP server software packages
- Comparison of file transfer protocols
- Curl-loader – FTP/S loading/testing open-source software
- DTXT
- File eXchange Protocol (FXP)
- File Service Protocol (FSP)
- FTAM
- FTPFS
- List of FTP commands

- List of FTP server return codes
- Managed File Transfer
- OBEX
- Shared file access
- TCP Wrapper

References

1. Forouzan, B.A. (2000). *TCP/IP: Protocol Suite* (1st ed.). New Delhi, India: Tata McGraw-Hill Publishing Company Limited.
2. Kozierok, Charles M. (2005). "The TCP/IP Guide v3.0" (http://www.tcpipguide.com/free/t_FTPOverviewHistoryandStandards.htm). Tcpipguide.com.
3. Dean, Tamara (2010). *Network+ Guide to Networks*. Delmar. pp. 168–171.
4. Vonau, Manuel (7 July 2021). "Firefox follows in Chrome's footsteps and drops FTP support (APK Download)" (https://www.androidpolice.com/2021/07/14/firefox-90-fully-removes-ftp-support-and-reorganizes-some-settings-apk-download/). *Android Police*. Retrieved 12 July 2021.
5. "Remove FTP support - Chrome Platform Status" (https://chromestatus.com/feature/6246151319715840). *www.chromestatus.com*. Retrieved 2 September 2021.
6. by, Written (23 March 2020). "Firefox is dropping FTP support" (https://news.sophos.com/en-us/2020/03/23/firefox-is-dropping-ftp-support/). *Sophos News*. Retrieved 13 October 2023.
7. Edwards, Benj (14 July 2022). "Chrome and Firefox Killed FTP Support: Here's an Easy Alternative" (https://www.howtogeek.com/744569/chrome-and-firefox-killed-ftp-support-heres-an-easy-alternative/). *How-To Geek*. Retrieved 13 October 2023.
8. Clark, M.P. (2003). *Data Networks IP and the Internet* (1st ed.). West Sussex, England: John Wiley & Sons Ltd.
9. "Active FTP vs. Passive FTP, a Definitive Explanation" (http://slacksite.com/other/ftp.html). Slacksite.com.
10. RFC 959 (https://www.rfc-editor.org/rfc/rfc959) (Standard) File Transfer Protocol (FTP). Postel, J. & Reynolds, J. (October 1985).
11. RFC 2428 (https://www.rfc-editor.org/rfc/rfc2428) (Proposed Standard) Extensions for IPv6, NAT, and Extended Passive Mode. Allman, M. & Metz, C. & Ostermann, S. (September 1998).
12. Stevens, W. Richard (1994). *TCP/IP Illustrated Volume I*. Vol. 1. Reading, Massachusetts, USA: Addison-Wesley Publishing Company. ISBN 0-201-63346-9.
13. Gleason, Mike (2005). "The File Transfer Protocol and Your Firewall/NAT" (https://www.ncftp.com/ncftpd/doc/misc/ftp_and_firewalls.html). Ncftp.com.
14. Klensin, John. *FTP TYPE Extension for Internationalized Text* (https://datatracker.ietf.org/doc/html/draft-klensin-ftpext-typeu-00). I-D draft-klensin-ftpext-typeu-00. Retrieved 9 June 2020.
15. Preston, J. (January 2005). *Deflate transmission mode for FTP* (https://datatracker.ietf.org/doc/html/draft-preston-ftpext-deflate-03). IETF. I-D draft-preston-ftpext-deflate-03. Retrieved 27 January 2016.
16. Allcock, W. (April 2003). "GridFTP: Protocol Extensions to FTP for the Grid" (https://ogf.org/documents/GFD.20.pdf) (PDF).
17. Mandrichenko, I. (4 May 2005). "GridFTP v2 Protocol Description" (https://ogf.org/documents/GFD.47.pdf) (PDF).
18. "MFMT FTP command" (https://support.solarwinds.com/SuccessCenter/s/article/MFMT-FTP-command). *support.solarwinds.com*. 11 October 2018.
19. "FTP Commands: DSIZ, MFCT, MFMT, AVBL, PASS, XPWD, XMKD | Serv-U" (https://www.serv-u.com/resources/tutorial/dsiz-mfct-mfmt-avbl-pass-xpwd-xmkd-ftp-command). *www.serv-u.com*.

20. "MDTM FTP command" (https://support.solarwinds.com/SuccessCenter/s/article/MDTM-FTP-command). *support.solarwinds.com*. 11 October 2018.
21. Prince, Brian (24 January 2012). "Should Organizations Retire FTP for Security?" (https://www.securityweek.com/should-organizations-retire-ftp-security/). *Security Week*. Retrieved 14 September 2017.
22. RFC 1635 (https://www.rfc-editor.org/rfc/rfc1635) (Informational) How to Use Anonymous FTP. P. & Emtage, A. & Marine, A. (May 1994).
23. FTP Access through Windows Explorer (https://helpdesk.egnyte.com/hc/en-us/articles/201637914-FTP-Access-through-Windows-Explorer)
24. "CSC373/406: SSH [2011/03/27-29]" (https://fpl.cs.depaul.edu/jriely/373/lectures/class-01-014.html). *fpl.cs.depaul.edu*. Retrieved 13 October 2023.
25. "FTP" (https://docs.kde.org/stable5/en/konqueror/konqueror/ftp.html). *docs.kde.org*. Retrieved 13 October 2023.
26. Cohen, Brent (26 July 2023). "How To Connect to FTP/SFTP in Dolphin | DeviceTests" (https://devicetests.com/connect-ftp-sftp-dolphin). Retrieved 13 October 2023.
27. Staff, Moyens (28 February 2022). "Samsung My Files vs Google Files: Which File Manager is Better on Galaxy Phones" (https://uk.moyens.net/android/samsung-my-files-vs-google-files-which-file-manager-is-better-on-galaxy-phones/). *Moyens I/O*. Retrieved 13 October 2023.
28. Matthews, J. (2005). *Computer Networking: Internet Protocols in Action* (1st ed.). Danvers, MA: John Wiley & Sons Inc.
29. Sneddon, Joey (26 January 2021). "Linux Release Roundup: GParted, Lightworks, Google Chrome + More" (https://www.omgubuntu.co.uk/2021/01/linux-release-roundup-chrome-lightworks-more). *omgubuntu.co.uk*. Retrieved 30 January 2021.
30. "See what's new in Firefox: 88.0 Firefox Release" (https://www.mozilla.org/en-US/firefox/88.0/releasenotes/). *mozilla.org*. 19 April 2021. Retrieved 20 April 2021.
31. "FireFTP - The Free FTP Client for Waterfox" (https://web.archive.org/web/20220301212656/https://fireftp.net/). *FireFTP.net*. Archived from the original (https://fireftp.net/) on 1 March 2022.
32. "URL Schemes Supported in Lynx" (https://lynx.invisible-island.net/lynx_help/lynx_url_support.html). *Lynx website*. Retrieved 6 July 2023.
33. "Accessing FTP servers | How to | Firefox Help" (https://support.mozilla.org/en-US/kb/Accessing%20FTP%20servers#w_ftp-servers-that-require-a-username-and-password). Support.mozilla.com. 5 September 2012. Retrieved 16 January 2013.
34. "How to Enter FTP Site Password in Internet Explorer" (https://web.archive.org/web/20150702005840/https://support.microsoft.com/en-us/kb/135975). Archived from the original (https://support.microsoft.com/en-us/kb/135975) on 2 July 2015. Retrieved 13 February 2020. Written for IE versions 6 and earlier. Might work with newer versions.
35. Jukka “Yucca” Korpela (18 September 1997). "FTP URLs" (https://jkorpela.fi/ftpurl.html). "IT and communication" (jkorpela.fi). Retrieved 26 January 2020.
36. "DownloadStudio - Internet Download Manager And Download Accelerator - Features" (http://www.conceiva.com/products/downloadstudio/features.asp). Conceiva. Retrieved 19 October 2021.
37. "LibreOffice 7.4: Release Notes" (https://wiki.documentfoundation.org/ReleaseNotes/7.4). The Document Foundation's Wiki. Retrieved 10 September 2022.
38. "ReleaseNotes/24.2" (https://wiki.documentfoundation.org/ReleaseNotes/24.2). The Document Foundation's Wiki. Retrieved 24 March 2024.
39. "Securing FTP using SSH" (https://nurdletech.com/linux-notes/ftp/ssh.html). Nurdletech.com.

40. "Components of the Information Assurance Platform (section Tectia ConnectSecure)" (https://web.archive.org/web/20200731160323/https://www.ssh.com/manuals/mft-events-product/63/ssh-solutions-your-business-components.html). *ssh.com*. Archived from the original (https://www.ssh.com/manuals/mft-events-product/63/ssh-solutions-your-business-components.html) on 31 July 2020.

Further reading

- RFC 697 (https://www.rfc-editor.org/rfc/rfc697) – CWD Command of FTP. July 1975.
- RFC 959 (https://www.rfc-editor.org/rfc/rfc959) – (Standard) File Transfer Protocol (FTP). J. Postel, J. Reynolds. October 1985.
- RFC 1579 (https://www.rfc-editor.org/rfc/rfc1579) – (Informational) Firewall-Friendly FTP. February 1994.
- RFC 1635 (https://www.rfc-editor.org/rfc/rfc1635) – (Informational) How to Use Anonymous FTP. May 1994.
- RFC 1639 (https://www.rfc-editor.org/rfc/rfc1639) – FTP Operation Over Big Address Records (FOOBAR). June 1994.
- RFC 1738 (https://www.rfc-editor.org/rfc/rfc1738) – Uniform Resource Locators (URL). December 1994.
- RFC 2228 (https://www.rfc-editor.org/rfc/rfc2228) – (Proposed Standard) FTP Security Extensions. October 1997.
- RFC 2389 (https://www.rfc-editor.org/rfc/rfc2389) – (Proposed Standard) Feature negotiation mechanism for the File Transfer Protocol. August 1998.
- RFC 2428 (https://www.rfc-editor.org/rfc/rfc2428) – (Proposed Standard) Extensions for IPv6, NAT, and Extended passive mode. September 1998.
- RFC 2577 (https://www.rfc-editor.org/rfc/rfc2577) – (Informational) FTP Security Considerations. May 1999.
- RFC 2640 (https://www.rfc-editor.org/rfc/rfc2640) – (Proposed Standard) Internationalization of the File Transfer Protocol. July 1999.
- RFC 3659 (https://www.rfc-editor.org/rfc/rfc3659) – (Proposed Standard) Extensions to FTP. P. Hethmon. March 2007.
- RFC 5797 (https://www.rfc-editor.org/rfc/rfc5797) – (Proposed Standard) FTP Command and Extension Registry. March 2010.
- RFC 7151 (https://www.rfc-editor.org/rfc/rfc7151) – (Proposed Standard) File Transfer Protocol HOST Command for Virtual Hosts. March 2014.
- IANA FTP Commands and Extensions registry (https://www.iana.org/assignments/ftp-commands-extensions/ftp-commands-extensions.xhtml) – The official registry of FTP Commands and Extensions

External links

- Communication Networks/File Transfer Protocol at Wikibooks
- FTP Server Online Tester (https://servertest.online/ftp) Authentication, encryption, mode and connectivity.
- Anonymous FTP Servers by Country Code TLD (2012): "Offbeat Internet - Public Access - FTP" (https://web.archive.org/web/20230328113445/https://www.jumpjet.info/Offbeat-Internet/Public/FTP/url.htm). *www.jumpjet.info*. 2012. Archived from the original (https://www.jumpjet.info/Offbeat-Internet/Public/FTP/url.htm) on 28 March 2023. Retrieved 16 January 2020.

Retrieved from "https://en.wikipedia.org/w/index.php?title=File_Transfer_Protocol&oldid=1292510762"